UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/659,864	09/12/2000	J. Leslie Vogel III	0044860.P2436	5866

7590 11/09/2006

Sheryl Sue Holloway
Blakely Sokoloff Taylor & Zafman LLP
12400 Wilshire Boulevard &th Floor
Los Angeles, CA 90025

EXAMINER	
TRAN, TONGOC	

ART UNIT	PAPER NUMBER
2134	

DATE MAILED: 11/09/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/659,864
Filing Date: September 12, 2000
Appellant(s): VOGEL, J. LESLIE

Sheryl Sue Holloway
Registration No. 37,850
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed July 17, 2006 appealing from the Office action mailed February 27, 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,526,506	LEWIS	2-2003
6178506 B1	QUICK, JR.	1-2001

Schneier, B., "Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C", 1996, pages 513-515.

ANSI/IEEE Std 802.11, 1999 Edition, "Information technology-Telecommunication and information exchange between systems--Local and metropolitan area networks--Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-3, 9-17, 19-22, 24-27, 29-32, 34-38, 40-48 and 50-51 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lewis (U.S. Patent No. 6,526,506) in view of Quick Jr. (U.S. Patent No. 6,178,506, hereinafter Quick)

In respect to claim 1, Lewis discloses a computerized method of establishing a secure wireless communications channel between an access point and a station, the channel being encrypted with a channel key, the method comprising:

sending, by the station to the access point through a setup connection, a request for a security preference for the access point (see Lewis, Fig. 6 and col. 10, line 46-col. 11, line 40);

sending, by the access point to the station through the setup connection, the security preference in response to the request when the access point can support the channel (see Lewis, col. 12, line 60-col. 13, line 15);

sending, by the station to the access point through the setup connection, the authentication information (see Lewis, col. 4, lines 27-42);

validating, by the access point, the station using the authentication information; encrypting, by the access point, the channel key using a second key when the station is validated (see Lewis, col. 4, lines 27-42 and col. 5, lines 29-41);

sending, by the access point to the station through the setup connection, the encrypted channel key (see Lewis, col. 5, lines 29-41);

decrypting, by the station, channel key in response to receiving the

encrypted channel key; and sending, by the station to the access point, data encrypted with the channel key to establish the channel (see Lewis, col. 5, line 10-col. 6, line 17).

Lewis discloses the mobile terminal sending authentication information (registering) with the access point (see Lewis, col. 4, lines 28-35) but does not explicitly discloses encrypting the authentication information. However, Quick discloses encrypting authentication information from mobile terminal to access point (Quick, col. 3, lines 1-10). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Quick's encrypting the authentication information with the teaching of Lewis' registering the mobile terminal with the access point in order to protect the user identification and password from compromise during the registration process (Quick, col. 2, lines 46-49).

In respect to claim 2, Lewis and Quick disclose the method of claim 1, wherein the first and second keys are a self-distributed key (see Quick, col. 4, line 45-col. 5, line 8).

In respect to claim 3, Lewis discloses the method of claim 1, Lewis wherein the first and second keys are a self distributed key and further comprising:

generating, by the access point, the self-distributed key using a security algorithm when the security preference is shared key; generating, by the station and sending to the access point, a first value using the security algorithm in response to receiving the security preference of shared key; generating, by the access point, and sending to the station, a second value using the security algorithm and the first value in response to receiving the first value; and calculating, by the station, the self-distributed key using the security algorithm and the second value in response to receiving the second value (see Quick, col. 4, line 45-col. 5, line 8).

In respect to claim 9, Lewis and Quick disclose the method of claim 2 further comprising:

encrypting, by the station, a name and password with the first key to generate the authentication information; and decrypting, by the access point, the name and password to validate the station (see Quick, col. 4, line 45-col. 5, line 8).

In respect to claim 10, Lewis and Quick disclose the method of claim 2 further comprising:

sending, by the access point to the station, a challenge; encrypting, by the station, the challenge with the first key to generate the authentication information; encrypting, by the access point, the challenge with the first key; and comparing,

by the access point, the authentication information with the challenge encrypted by the access point with the first key to validate the station (see Quick, col. 4, line 45-col. 5, line 8)

In respect to claim 11, Lewis and Quick disclose the method of claim 1, wherein the first key is a public key of a public-private key pair for the access point, and the second key is a public key of a public-private key pair for the station (see Quick, col. 4, line 45 -col. 5, line 8).

In respect to claim 12, Lewis and Quick disclose the method of claim 11 further comprising:

sending, by the access point to the station, the first key; and.

sending, by the station to the access point, the second key (see Quick col. 4, line 45-col. 5, line 8)

In respect to claim 13, Lewis and Quick disclose the method of claim 12, wherein the second key is sent to the access point when the request for the security preference is sent by the station (see Quick, col. 4, line 45-col. 5, line 8).

In respect to claim 14, Lewis and Quick disclose the method of claim 12, wherein the first key is sent to the station when the security preference is sent by the access point (see Quick, col. 4, line 45-col. 5, line 8).

In respect to claim 15, Lewis discloses the method of claim 1, wherein establishing the channel creates a standard wired equivalent privacy (WEP) network, and the station and the access point exchange messages conforming to a format required by the standard that defines a WEP network to establish the WEP network (see Lewis, col. 2, lines 18-43).

In respect to claim 16, 21, 26, 31 and 36-37, 40, 42-47 and 50, the claim limitations are substantially similar to claim 1. Therefore, claims 16, 21, 26, 31, 36-37, 40, 42-47 and 50 are rejected based on the similar rationale.

In respect to claim 17, the claim limitation is substantially similar to claim 3. Therefore, claim 17 is rejected based on the similar rationale.

In respect to claim 19, the method of claim 16 further comprising:

using a first key to generate the authentication information; and

using a second key to decrypt the encrypted channel key (see Lewis, col. 5, line 10-col. 6, line 17).

In respect to claims 20, 25, 30, 35, 41 and 51, the claim limitations are substantially similar to claim 11. Therefore, claims 20, 25, 30 and 35 are rejected based on the similar rationale.

In respect to claims 24, 29 and 34, the claim limitations are substantially similar to claim 19. Therefore, claims 24, 29 and 34 are rejected based on the similar rationale.

In respect to claim 22, the claim limitation is substantially similar to claim 3. Therefore, claim 22 is rejected based on the similar rationale.

In respect to claim 27, the claim limitation is substantially similar to claim 17. Therefore claim 27 is rejected based on the similar rationale.

In respect to claim 32, the claim limitation is substantially similar to claim 22. Therefore, claim 32 is rejected based on the similar rationale.

In respect to claim 38, Lewis and Quick disclose the secure wireless network of claim 37, wherein access point if further operable for encrypting the shared channel key using a self-distributed key for sending to the station and the station is further operable for decrypting the shared channel key upon receipt (see Quick, col. 4, line 45-col. 5, line 8).

In respect to claim 48, the claim limitation is substantially similar to claim 38. Therefore, claim 48 is rejected based on the similar rationale.

2. Claims 4-8, 18, 23, 28, 33, 39 and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lewis (U.S. Patent No. 6,526,506) in view of Quick Jr. (U.S. Patent No. 6,178,506, hereinafter Quick) and further in view of Schneier ("Applied Cryptography, Second Edition, Protocols, Algorithms, and Source Code in C", John Wiley & Sons, Inc., 1996, hereinafter Schneier).

In respect to claim 4, Lewis and Quick disclose the method of claim 3. Lewis and Quick do not disclose but Schneier discloses wherein the security algorithm g^n mod p and further comprising: obtaining, by the access point, integers x, g and p to generate the self-distributed key $k = g^x \bmod p$ (Schneier, page 515, Hughes, (1) Alice (access point) chooses a random large integer x and generates $K=g^x$ mode n (or p)); obtaining, by the station , the integers g and p, and an integer y to generate the first value $Y = g^y \bmod p$ (Schneier, page 515, Hughes, (2) Bob (or station) choose a random large ineger y and sends Alice, $Y=g^y$ mod n (or p); generating, by the access point, the second value $X = Y^x \bmod p$ (Schneier, page 515, Hughes, (3) Alice sends Bob, $X = Y^x$ mod n (or p); and setting, by the station, z equal to y^{-1} to calculate the self-distributed key $k = X^z \bmod p$ (see Schneier, page 515, Hughes, (4) Bob computes $z=y^{-1}$, $k' = X^z$ mod n (or p), If everything goes correctly, k=k'). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Schneier with the teaching of Lewis's wireless communication between mobile and access point and Quick's Diffie-Hellman's protocol with Schneier's teaching

of Hughes' protocol so that key can be computed before any interaction between the mobile station and the access point (see Schneier, page 515, Hughes).

In respect to claim 5, Lewis, Quick and Schneier disclose the method of claim 4 wherein obtaining, by the station, the integers g and p comprises:

sending, by the access point (Bob) to the station (Alice), the integers for g and p (see Schneier, page 515, g and n).

In respect to claim 6, Lewis, Quick and Schneier disclose the method of claim 5, wherein the integers for g and p (g and n) are sent to the station (Alice) when the security preferences are sent by the access point (Bob) (see Schneier, page 515, Hughes).

In respect to claim 7, Lewis, Quick and Schneier disclose the method of claim 5, wherein g and p are sent to the station when a user name and password for the station are registered with the access point (see Quick, col. 4, line 60 to col. 5, line 8).

In respect to claim 8, Lewis, Quick and Schneier discloses the method of claim 4 further comprising:

publishing, by the access point, the integers g and p for a set of stations (see Schneier, page 515).

In respect to claims 18, 23, 28 and 33, the claim limitations are substantially similar to claim 4. Therefore, claims 18, 23, 28 and 33 are rejected based on the similar rationale.

In respect to claim 39, Lewis and Quick disclose the secure wireless network of claim 38. Lewis and Quick do not disclose but Schneier discloses wherein the station and the access point are further operable for calculating the self-distributed key by exchanging messages in accordance with the Hughes transmission protocol (see Schneier, page 515, Hughes). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Schneier with the teaching of Lewis's wireless communication between mobile and access point and Quick's Diffie-Hellman's protocol with Schneier's teaching of Hughes' protocol so that key can be computed before any interaction between the mobile station and the access point (see Schneier, page 515, Hughes and Key Exchange Without Exchanging Keys).

In respect to claim 49, the claim limitation is substantially similar to claim 39. Therefore, claim 49 is rejected based on the similar rationale.

(10) Response to Argument

Appellant's remarks in the Brief on pages 3-4, with respect to 35 U.S.C. 112 claim rejection where the Examiner rejected the amended limitation for failing to provide

support in the Specification. The amended limitation recites "wherein the security preference specifies one authentication protocol from a set of authentication protocols supported by the access point". In the Brief, Appellant refers to page 10, line 20 through page 11 of Appellant's Specification for support by sets forth one example of a security preference as being "shared key" and page 19, lines 1-5 sets forth another example for another types of authentication for wireless networks as "open system". Therefore, claims 1, 16, 21, 26, 31, 36, 42 and 46 rejected under 35 U.S.C. 112 are now withdrawn.

Appellant's remarks in the Brief on page 4 with respect to 35 U.S.C. 102(a) rejection over cited prior art Patiyoot where two types of authentication protocol are proposed to be used in wireless ATM network. Patiyoot describes the different mechanisms for the authentication of wireless ATM networks using both secret and public keys. However, upon reconsidering the art, Examiner agrees with the Appellant's remark in the Brief that Patiyoot does not teach or suggest an access point sends a security preference that is one of a set of authentication protocols supported by the access point as claimed in claim 1. Therefore, claims 1, 16, 21, 31, 42 and 46 rejected under 35 U.S.C. 102(a) are now withdrawn.

Appellant's remarks in the Brief on pages 5-7 with respect to 35 U.S.C. 103(a) rejection over the combination of Lewis and Quick where Appellant argues that the cited prior art fail to teach all the claimed limitations. Appellant's remark in the Brief has been

fully considered in light of the Specification and the cited prior art with the support from the Appendix submitted with the Brief, the section 8.1 f IEEE 802.11 standard (hereinafter Appendix), where "open system" and "shared key" are provided with more detailed description.

Response to Appellant's remarks in the Brief on page 5:

Appellant argues that both Lewis and Quick teach only a single authentication protocol (Brief, page 5, In Lewis, "i.e., the shared network encryption key"; In Quick, "the home server shares a public/private key pair with the mobile device while the local server shared an authentication key with the mobile device).

In the Specification on page 19, Appellant describes "[t]he invention is particularly suited for use with Intrastructure Networks defined by the 802.11 standard" and "[t]he AP specifies whether access to the LAN is open to all stations ("Open System") or secured through a Wired Equipvalent Privacy (EP) protocol using a shared key and a WEP encryption algorithm ("Shared Key"). Similarly, Lewis teaches a wireless network authentication in accordance with the IEEE 802.11 standard for "basic" mobile terminal (BMT) to access network without engaging in secure encrypted communication and mobile terminal with encrypted communication (i.e. col. 5, lines 10-27, col. 15, line 54-col. 6, line 5, BMT, WEP protocol in IEEE 802.11). According to the Appendix, IEEE 802.11 defines two subtypes of authentication service: Open System and Shared Key. Open System authentication is the default authentication protocol for 802.11, it authenticates anyone who requests it and provides a null authentication process

whereas the Shared key authentication uses a standard challenge and response along with a shared secret key to provide authentication (Appendix, pages 59-61). Therefore, the wireless communication between the mobile terminal and the access point taught by Lewis clearly encompasses at least one authentication protocol in accordance with the 802.11 standard as claimed.

Response to Appellant's remark in the Brief on page 6:

Appellant argues that both Lewis and Quick fail to teach "generating authentication information using a key". In the Brief, Appellant disagrees with the Examiner in the Final Rejection that authentication information obtained during the registration with the access point taught by Lewis or the encrypted public key taught by Quick meet the claimed limitation. Appellant asserts that the claimed language is "generating the authentication information using a key" and not "encrypting the authentication using a key". However, in the Specification, pages 4 and 5, Appellant describes "[t]he authentication information can be a user name and password, an encrypted challenge such as used in the challenge Handshake Authentication Protocol, or other types of data typical used to authenticate clients on a network. In one aspect, the first and second keys are identical keys. In another aspect, the first key is a public key for the access point and the second key is a public key for the station". In Lewis, the authentication information including information provided through registration (col. 4, lines 33-35, "conventional techniques", i.e. device ID, network address, user name or pin); or shared key authentication exchanged (col. 6, lines 50-51, WEP protocol in IEEE

802.11 standard or Appendix, pages 60-61, challenge or encrypted challenge). Appellant argues that Quick's authentication information includes a public key, but Quick does not teach or suggest that the public key is generated using a key as claimed. However, in col. 4, lines 45-60, Quick teaches a public key is concatenated with a random number, encrypts the concatenated number with a password using H-EKE. A key is essentially a random number, by concatenating the random number with the public key. Therefore, Quick meets the claimed limitation of generating the authentication information using a key.

Appellant further argues on page 6 in the Brief that Lewis only discloses the exchange of network encryption keys not security preferences as defined by Appellant. Appellant equates security preferences as authentication protocol. Since Lewis teaches authentication in accordance with IEEE 802.11 which defines Open System and Shared Key. According to the Appendix, each frame encompasses Authentication Algorithm Identification indicating whether it is Open System or Shared Key (Appendix, pages 59-61). This support from the Appendix is in line with Appellant's disclosure where Appellant describes after receiving request from the user station for connection to the AP. If the AP can handle a new connection, it sends its security preference, i.e. shared key, to the user station (Specification, page 11).

Appellant further argues on page 6 that Lewis does not teach or suggest that the mobile device receives a new encryption key from the access point in response to the mobile device requesting the key. However, in one embodiment, Lewis teaches the mobile terminal is always checking to determine if a new encrypt key has been received

from the access point and if it receives the new encrypt key from the access point, it provides the encrypt key to the encryption engine (col. 12, lines 49-59). In another embodiment, in response to the mobile terminal's "GET KEY" request, the access point transmits the key to the mobile terminal (col. 18, lines 9-18, the key distribution server "responds to the request by transmitting the encrypt key to the requesting mobile terminal *via the access point*" meets the claimed limitation of "sendng, by the access point, the channel key through the setup connection, the encrypted channel key".

In response to Appellant's argument in the Brief on page 7:

Appellant argues that neither Lewis nor Quick disclose any data structure as claimed in claim 42-45. Lewis does teach data structure as indicated in Fig. 4 and 5. In Fig. 4, Lewis teaches a system device table with different fields representing Authorized Device ID, Non-Encrypt Acess or Access Expirations. In Fig. 5, Lewis teaches a Clear Table with a field representing the Device ID.

In response to Appellant's argument in the Brief on page 8 in respect to 35 U.S.C. 103(a) rejection over the combination of Lewis, Quick and Schneier.

Appellant argues that claim 4 which is the representative claim for the 103(a) rejection and claim a particular security algorithm that is used to generate a key for the access point. Appellant argues that the Examiner has failed to state a proper prima facie case of obviousness because the combination of Lewis, Quick and Schneier does not teach each and every limitation of Appellant's claim 4. Appellant argues since claim

4 includes all the limitation of claim 1, at least one of the references must disclose an access point that sends a security preference as claimed in claim 1. As previously stated, Appellant equates security preferences as authentication protocol. Since Lewis teaches authentication in accordance with IEEE 802.11 which defines Open System and Shared Key. According to the Appendix, each frame encompasses Authentication Algorithm Identification indicating whether it is Open System or Shared Key (Appendix, pages 59-61). This support from the Appendix is in line with Appellant's disclosure where Appellant describes after receiving request from the user station for connection to the AP, if the AP can handle a new connection, it sends its security preference, i.e. shared key, to the user station (Specification, page 11).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

GILBERTO BARRON JR
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Respectfully submitted,

Tongoc Tran
Examiner
Art Unit 2134

Conferees:

Gilberto Barron 

Benjamin Lanier

ANSI/IEEE Std 802.11, 1999 Edition

Information technology— Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements—

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

Sponsor

LAN MAN Standards Committee
of the
IEEE Computer Society

ANSI/IEEE Std 802.11, 1999 Edition

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying all patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

The patent holder has, however, filed a statement of assurance that it will grant a license under these rights without compensation or under reasonable rates and nondiscriminatory, reasonable terms and conditions to all applicants desiring to obtain such a license. The IEEE makes no representation as to the reasonableness of rates and/or terms and conditions of the license agreement offered by the patent holder. Contact information may be obtained from the IEEE Standards Department.

Contents

8. Authentication and privacy

8.1 Authentication services

IEEE 802.11 defines two subtypes of authentication service: *Open System* and *Shared Key*. The subtype invoked is indicated in the body of authentication management frames. Thus authentication frames are self-identifying with respect to authentication algorithm. All management frames of subtype Authentication shall be unicast frames as authentication is performed between pairs of stations (i.e., multicast authentication is not allowed). Management frames of subtype Deauthentication are advisory, and may therefore be sent as group-addressed frames.

A mutual authentication relationship shall exist between two stations following a successful authentication exchange as described below. Authentication shall be used between stations and the AP in an infrastructure BSS. Authentication may be used between two STAs in an IBSS.

8.1.1 Open System authentication

Open System authentication is the simplest of the available authentication algorithms. Essentially it is a null authentication algorithm. Any STA that requests authentication with this algorithm may become authenticated if dot11AuthenticationType at the recipient station is set to Open System authentication. Open System authentication is not required to be successful as a STA may decline to authenticate with any particular other STA. Open System authentication is the default authentication algorithm.

Open System authentication involves a two-step authentication transaction sequence. The first step in the sequence is the identity assertion and request for authentication. The second step in the sequence is the authentication result. If the result is "successful," the STAs shall be mutually authenticated.

8.1.1.1 Open System authentication (first frame)

— Message type: Management
— Message subtype: Authentication
— Information items:
  • Authentication Algorithm Identification = "Open System"
  • Station Identity Assertion (in SA field of header)
  • Authentication transaction sequence number = 1
  • Authentication algorithm dependent information (none)
— Direction of message: From authentication initiating STA to authenticating STA

8.1.1.2 Open System authentication (final frame)

— Message type: Management
— Message subtype: Authentication
— Information items:
  • Authentication Algorithm Identification = "Open System"
  • Authentication transaction sequence number = 2
  • Authentication algorithm dependent information (none)
  • The result of the requested authentication as defined in 7.3.1.9
— Direction of message: From authenticating STA to initiating STA

If dot11AuthenticationType does not include the value "Open System," the result code shall not take the value "successful."

8.1.2 Shared Key authentication

Shared Key authentication supports authentication of STAs as either a member of those who know a shared secret key or a member of those who do not. IEEE 802.11 Shared Key authentication accomplishes this without the need to transmit the secret key in the clear; however, it does require the use of the WEP privacy mechanism. Therefore, this authentication scheme is only available if the WEP option is implemented. Additionally, the Shared Key authentication algorithm shall be implemented as one of the dot11AuthenticationAlgorithms at any STA where WEP is implemented.

The required secret, shared key is presumed to have been delivered to participating STAs via a secure channel that is independent of IEEE 802.11. This shared key is contained in a write-only MIB attribute via the MAC management path. The attribute is write-only so that the key value remains internal to the MAC.

During the Shared Key authentication exchange, both the challenge and the encrypted challenge are transmitted. This facilitates unauthorized discovery of the pseudorandom number (PRN) sequence for the key/IV pair used for the exchange. Implementations should therefore avoid using the same key/IV pair for subsequent frames.

A STA shall not initiate a Shared Key authentication exchange unless its dot11PrivacyOptionImplemented attribute is "true."

In the following description, the STA initiating the authentication exchange is referred to as the *requester*, and the STA to which the initial frame in the exchange is addressed is referred to as the *responder*.

8.1.2.1 Shared Key authentication (first frame)

— Message type: Management
— Message subtype: Authentication
— Information Items:
  • Station Identity Assertion (in SA field of header)
  • Authentication Algorithm Identification = "Shared Key"
  • Authentication transaction sequence number = 1
  • Authentication algorithm dependent information (none)
— Direction of message: From requester to responder

8.1.2.2 Shared Key authentication (second frame)

Before sending the second frame in the Shared Key authentication sequence, the responder shall use WEP to generate a string of octets that shall be used as the authentication challenge text.

— Message type: Management
— Message subtype: Authentication
— Information Items:
  • Authentication Algorithm Identification = "Shared Key"
  • Authentication transaction sequence number = 2
  • Authentication algorithm dependent information = the authentication result.
  • The result of the requested authentication as defined in 7.3.1.9

    If the status code is not "successful," this shall be the last frame of the transaction sequence. If the status code is not "successful," the content of the challenge text field is unspecified.

    If the status code is "successful," the following additional information items shall have valid contents:

    Authentication algorithm dependent information = challenge text.

This field shall be of fixed length of 128 octets. The field shall be filled with octets generated by the WEP pseudo-random number generator (PRNG). The actual value of the challenge field is unimportant, but the value shall not be a single static value. The key and IV used when generating the challenge text are unspecified because this key/IV value does not have to be shared and does not affect interoperability.

— Direction of message: From responder to requester

8.1.2.3 Shared Key authentication (third frame)

The requester shall copy the challenge text from the second frame into the third frame. The third frame shall be transmitted after encryption by WEP, as defined in 8.2.3, using the shared secret key.

— Message type: Management
— Message subtype: Authentication
— Information Items:
  • Authentication Algorithm Identification = "Shared Key"
  • Authentication transaction sequence number = 3
  • Authentication algorithm dependent information = challenge text from sequence two frame
— Direction of message: From requester to responder

This frame shall be encrypted as described below.

8.1.2.4 Shared Key authentication (final frame)

The responder shall attempt to decrypt the contents of the third frame in the authentication sequence as described below. If the WEP ICV check is successful, the responder shall then compare the decrypted contents of the Challenge Text field to the challenge text that was sent in Frame 2 of the sequence. If they are the same, then the responder shall respond with a successful status code in Frame 4 of the sequence. If the WEP ICV check fails, the responder shall respond with an unsuccessful status code in Frame 4 of the sequence as described below.

— Message type: Management
— Message subtype: Authentication
— Information Items:
  • Authentication Algorithm Identification = "Shared Key"
  • Authentication transaction sequence number = 4
  • Authentication algorithm dependent information = the authentication result
    The result of the requested authentication.
    This is a fixed length item with values "successful" and "unsuccessful."
— Direction of message: From responder to requester

8.2 The Wired Equivalent Privacy (WEP) algorithm

8.2.1 Introduction

Eavesdropping is a familiar problem to users of other types of wireless technology. IEEE 802.11 specifies a wired LAN equivalent data confidentiality algorithm. *Wired equivalent privacy* is defined as protecting authorized users of a wireless LAN from casual eavesdropping. This service is intended to provide functionality for the wireless LAN equivalent to that provided by the physical security attributes inherent to a wired medium.

Data confidentiality depends on an external key management service to distribute data enciphering/deciphering keys. The IEEE 802.11 standards committee specifically recommends against running an IEEE 802.11

RELATED PROCEEDINGS APPENDIX FOR APPEAL BRIEF UNDER 37 C.F.R. § 41.37

NONE

TRANSMITTAL **PATENT**

Application No.: 09/659,864
Filing Date: September 12, 2000
First Named Inventor J. Leslie Vogel, III
Examiner's Name: Tongoc Tran
Art Unit: 2134
Attorney Docket No.: 004860.P2436

____ An Amendment After Final Action (37 CFR 1.116) is attached and applicant(s) request expedited action.
__X__ Charge any fee not covered by any check submitted to Deposit Account No. 02-2666.
__X__ Applicant(s) hereby request and authorize the U.S. Patent and Trademark Office to (1) treat any concurrent or future reply that requires a petition for extension of time as incorporating a petition for extension of time for the appropriate length of time and (2) charge all required fees, including extension of time fees and fees under 37 CFR 1.16 and 1.17, for any concurrent or future reply to Deposit Account No. 02-2666.
____ Applicant(s) claim small entity status (37 CFR 1.27).

ATTACHMENTS

____ Preliminary Amendment
____ Amendment/Response with respect to Office Action
____ Amendment/Response After Final Action (37 CFR 1.116) (reminder: consider filing a Notice of Appeal)
____ Notice of Appeal
____ RCE (Request for Continued Examination)
____ Supplemental Declaration
____ Terminal Disclaimer (reminder: if executed by an attorney, the attorney must be properly of record)
____ Information Disclosure Statement (IDS)
____ Copies of IDS citations
____ Petition for Extension of Time
__X__ Fee Transmittal Document (that includes a fee calculation based on the type and number of claims)
____ Cross-Reference to Related Application(s)
____ Certified Copy of Priority Document
__X__ Other: Appeal Brief
__X__ Other: Exhibit A-13 pages, Exhibit B-12 pages, Exhibit C- 1 page
__X__ Check(s)
__X__ Postcard (Return Receipt)

SUBMITTED BY:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
TYPED OR PRINTED NAME: Sheryl Sue Holloway
SIGNATURE:
REG. NO.: 37,850
DATE: July 17, 2006
ADDRESS: 12400 Wilshire Boulevard, Seventh Floor
Los Angeles, California 90025
TELEPHONE NO.: (408) 720-8300

CERTIFICATE OF MAILING BY FIRST CLASS MAIL (if applicable)

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria Virginia 22313-1450
on 7.17.06
Date of Deposit
Carla Anysia Nascimento
Name of Person Mailing Correspondence
Signature
7.17.06
Date

Express Mail Label No. (if applicable):

(10/14/03)

Send to: COMMISSIONER FOR PATENTS, P.O. Box 1450, Alexandria, Virginia 22313-1450

 AFA

FEE TRANSMITTAL FOR FY 2006

Effective on 12/08/2004. Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

TOTAL AMOUNT OF PAYMENT ($) 500.00

Complete if Known:

Application No.	09/659,864
Filing Date	September 12, 2000
First Named Inventor	J. Leslie Vogel, III
Examiner Name	Tongoc Tran
Art Unit	2134
Attorney Docket No.	004860.P2436

_____ Applicant claims small entity status. See 37 CFR 1.27.

METHOD OF PAYMENT (check all that apply)

_X_ Check _____ Credit Card _____ Money Order ____ None ____ Other (please identify)

_____ Deposit Account

Deposit Account Number : 02-2666

Deposit Account Name: _____

_X_ The Director is Authorized to do the following with respect to the above-identified Deposit Account:

- ____ Charge fee(s) indicated below.
- _X_ Charge any additional fee(s) or underpayment of fee(s) during the pendency of this application.
- ____ Charge fee(s) indicated below except for the filing fee
- _X_ Credit any overpayments.
- _X_ Any concurrent or future reply that requires a petition for extension of time should be treated as incorporating an appropriate petition for extension of time and all required fees should be charged.

Warning: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description		Fees Paid ($)
1011	300	2011	150	Utility application filing fee		
1111	500	2111	250	Utility search fee	1,000/500	
1311	200	2311	100	Utility examination fee		
1012	200	2012	100	Design application filing fee		
1112	100	2112	50	Design search fee	430/215	
1312	130	2312	65	Design examination fee		
1013	200	2013	100	Plant filing fee		
1113	300	2113	150	Plant search fee	660/330	
1313	160	2313	80	Plant examination fee		
1004	300	2004	150	Reissue filing fee		
1114	500	2114	250	Reissue search fee	1,400/700	
1314	600	2314	300	Reissue examination fee		
1005	200	2005	100	Provisional application filing fee		

SUBTOTAL (1) $ 0.00

2. EXCESS CLAIM FEES

	Extra Claims		Fee from below		Fees Paid ($)
Total Claims	_____ – 20 or HP =	_____	X $50.00	=	_____
HP = highest number of total claims paid for, if greater than 20					
Independent Claims	_____ – 3 or HP =	_____	X $200.00	=	_____
HP = highest number of independent claims paid for, if greater than 3					
Multiple Dependent Claims			_____	=	_____

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	50	2202	25	Each claim over 20
1201	200	2201	100	Each independent claim over 3
1203	360	2203	180	Multiple dependent claims, if not paid
1204	200	2204	100	Reissue: each claim over 20 and more than in the original patent
1205	50	2205	25	Reissue: each independent claim more than in the original patent

SUBTOTAL (2) $ 0.00

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets		Extra Sheets		Number of each add'l 50 or fraction thereof		Fee from below		Fees paid ($)
_____	– 100 =	_____	/ 50 =	_____ (round up to whole number)	X	$250.00		_____

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description: Application size fee for each additional group of 50 sheets beyond initial 100 sheets (count spec & drawings except sequences & program listings):
1081	250	2081	125	Utility
1082	250	2082	125	Design
1083	250	2083	125	Plant
1084	250	2084	125	Reissue

SUBTOTAL (3) $ 0.00

FEE CALCULATION (continued)

4. OTHER FEE(S)

Non-English Specification, $130 fee (no small entity discount) — Fees Paid ($): _____

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fees Paid ($)
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for ex parte reexamination	
1813	8,800	1813	8,800	Request for inter parties reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	120	2251	60	Extension for reply within first month	
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	
1254	1,590	2254	795	Extension for reply within fourth month	
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a brief in support of an appeal	$500.00
1403	1,000	2403	500	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	500	2452	250	Petition to revive – unavoidable	
1453	1,500	2453	750	Petition to revive - unintentional	
1501	1,400	2501	700	Utility issue fee (or reissue)	
1502	800	2502	400	Design issue fee	
1503	1100	2503	550	Plant issue fee	
1462	400	1462	400	Petitions to the Commissioner (CFR 1.17(f) Group I)	
1463	200	1463	200	Petitions to the Commissioner (CFR 1.17(g) Group II)	
1464	130	1464	130	Petitions to the Commissioner (CFR 1.17(h) Group III)	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	For filing a submission after final rejection (see 37 CFR 1.129(a))	
1814	130	2814	65	Statutory Disclaimer	
1810	790	2810	395	For each additional invention to be examined (see 37 CFR 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	
1504	300	1504	300	Publication fee for early, voluntary, or normal pub.	
1505	300	1505	300	Publication fee for republication	
1803	130	1803	130	Request for voluntary publication or republication	
1808	130	1808	130	Processing fee under 37 CFR 1.17(i) (except provisionals)	
1454	1,370	1454	1,370	Acceptance of unintentionally delayed claim for priority	

Other fee (specify) _____ _____

Other fee (specify) _____ _____

SUBTOTAL (4) $ 500.00

*Reduced by Basic Filing Fee Paid

SUBMITTED BY:

Typed or Printed Name: Sheryl Sue Holloway

Signature: _____ Date: July 17, 2006

Reg. Number: 37,850 Telephone Number: 408-720-8300

Send to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

APPLIED CRYPTOGRAPHY, SECOND EDITION

PROTOCOLS, ALGORITHMS, AND SOURCE CODE IN C

John Wiley & Sons, Inc.
New York • Chichester • Brisbane • Toronto • Singapore

Publisher: Katherine Schowalter
Editor: Phil Sutherland
Assistant Editor: Allison Roarty
Managing Editor: Robert Aronds
Text Design & Composition: North Market Street Graphics

This text is printed on acid-free paper.

Published by John Wiley & Sons, Inc.

Library of Congress Cataloging-in-Publication Data:

Schneier, Bruce
Applied Cryptography Second Edition : protocols, algorithms, and source code in C / Bruce Schneier.
p. cm.
Includes bibliographical references (p. 675).
ISBN 0-471-12845-7 (cloth : acid-free paper). — ISBN 0-471-11709-9 (paper : acid-free paper)
1. Computer security. 2. Telecommunication—Security measures. 3. Cryptography. I. Title.
QA76.9.A25S35 1996
005.8'2—dc20 95-12398
CIP

Printed in the United States of America
10 9 8 7 6

CHAPTER 1

Foundations

1.1 TERMINOLOGY

Sender and Receiver

Suppose a sender wants to send a message to a receiver. Moreover, this sender wants to send the message securely: She wants to make sure an eavesdropper cannot read the message.

Messages and Encryption

A message is **plaintext** (sometimes called cleartext). The process of disguising a message in such a way as to hide its substance is **encryption**. An encrypted message is **ciphertext**. The process of turning ciphertext back into plaintext is **decryption**. This is all shown in Figure 1.1.

(If you want to follow the ISO 7498-2 standard, use the terms "encipher" and "decipher." It seems that some cultures find the terms "encrypt" and "decrypt" offensive, as they refer to dead bodies.)

The art and science of keeping messages secure is **cryptography**, and it is practiced by **cryptographers**. **Cryptanalysts** are practitioners of **cryptanalysis**, the art and science of breaking ciphertext; that is, seeing through the disguise. The branch of mathematics encompassing both cryptography and cryptanalysis is **cryptology** and its practitioners are **cryptologists**. Modern cryptologists are generally trained in theoretical mathematics—they have to be.

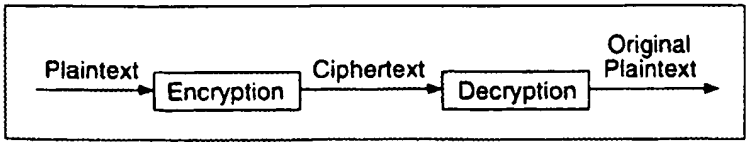

Figure 1.1 Encryption and Decryption.

Plaintext is denoted by *M*, for message, or *P*, for plaintext. It can be a stream of bits, a text file, a bitmap, a stream of digitized voice, a digital video image . . . whatever. As far as a computer is concerned, *M* is simply binary data. (After this chapter, this book concerns itself with binary data and computer cryptography.) The plaintext can be intended for either transmission or storage. In any case, *M* is the message to be encrypted.

Ciphertext is denoted by *C*. It is also binary data: sometimes the same size as *M*, sometimes larger. (By combining encryption with compression, *C* may be smaller than *M*. However, encryption does not accomplish this.) The encryption function *E*, operates on *M* to produce *C*. Or, in mathematical notation:

$$E(M) = C$$

In the reverse process, the decryption function *D* operates on *C* to produce *M*:

$$D(C) = M$$

Since the whole point of encrypting and then decrypting a message is to recover the original plaintext, the following identity must hold true:

$$D(E(M)) = M$$

Authentication, Integrity, and Nonrepudiation

In addition to providing confidentiality, cryptography is often asked to do other jobs:

- **Authentication.** It should be possible for the receiver of a message to ascertain its origin; an intruder should not be able to masquerade as someone else.
- **Integrity.** It should be possible for the receiver of a message to verify that it has not been modified in transit; an intruder should not be able to substitute a false message for a legitimate one.
- **Nonrepudiation.** A sender should not be able to falsely deny later that he sent a message.

These are vital requirements for social interaction on computers, and are analogous to face-to-face interactions. That someone is who he says he is . . . that someone's credentials—whether a driver's license, a medical degree, or a passport—are valid . . . that a document purporting to come from a person actually came from that person. . . . These are the things that authentication, integrity, and nonrepudiation provide.

Algorithms and Keys

A **cryptographic algorithm**, also called a **cipher**, is the mathematical function used for encryption and decryption. (Generally, there are two related functions: one for encryption and the other for decryption.)

If the security of an algorithm is based on keeping the way that algorithm works a secret, it is a **restricted** algorithm. Restricted algorithms have historical interest, but are woefully inadequate by today's standards. A large or changing group of users cannot use them, because every time a user leaves the group everyone else must switch to a different algorithm. If someone accidentally reveals the secret, everyone must change their algorithm.

Even more damning, restricted algorithms allow no quality control or standardization. Every group of users must have their own unique algorithm. Such a group can't use off-the-shelf hardware or software products; an eavesdropper can buy the same product and learn the algorithm. They have to write their own algorithms and implementations. If no one in the group is a good cryptographer, then they won't know if they have a secure algorithm.

Despite these major drawbacks, restricted algorithms are enormously popular for low-security applications. Users either don't realize or don't care about the security problems inherent in their system.

Modern cryptography solves this problem with a **key**, denoted by K. This key might be any one of a large number of values. The range of possible values of the key is called the **keyspace**. Both the encryption and decryption operations use this key (i.e., they are dependent on the key and this fact is denoted by the K subscript), so the functions now become:

$$E_K(M) = C$$
$$D_K(C) = M$$

Those functions have the property that (see Figure 1.2):

$$D_K(E_K(M)) = M$$

Some algorithms use a different encryption key and decryption key (see Figure 1.3). That is, the encryption key, K_1, is different from the corresponding decryption key, K_2. In this case:

$$E_{K_1}(M) = C$$
$$D_{K_2}(C) = M$$
$$D_{K_2}(E_{K_1}(M)) = M$$

All of the security in these algorithms is based in the key (or keys); none is based in the details of the algorithm. This means that the algorithm can be published and analyzed. Products using the algorithm can be mass-produced. It doesn't matter if an

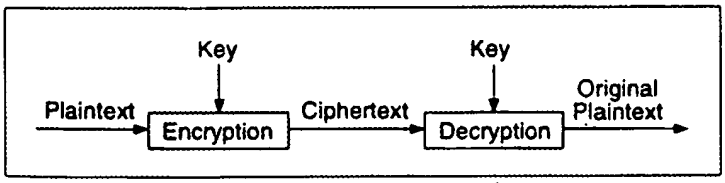

Figure 1.2 Encryption and decryption with a key.

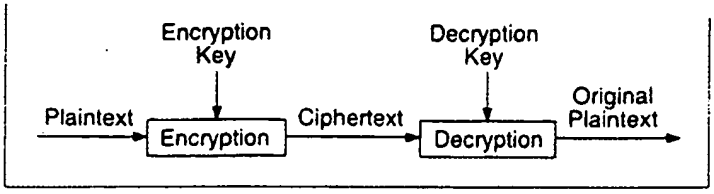

Figure 1.3 Encryption and decryption with two different keys.

eavesdropper knows your algorithm; if she doesn't know your particular key, she can't read your messages.

A **cryptosystem** is an algorithm, plus all possible plaintexts, ciphertexts, and keys.

Symmetric Algorithms

There are two general types of key-based algorithms: symmetric and public-key. **Symmetric algorithms**, sometimes called conventional algorithms, are algorithms where the encryption key can be calculated from the decryption key and vice versa. In most symmetric algorithms, the encryption key and the decryption key are the same. These algorithms, also called secret-key algorithms, single-key algorithms, or one-key algorithms, require that the sender and receiver agree on a key before they can communicate securely. The security of a symmetric algorithm rests in the key; divulging the key means that anyone could encrypt and decrypt messages. As long as the communication needs to remain secret, the key must remain secret.

Encryption and decryption with a symmetric algorithm are denoted by:

$$E_K(M) = C$$
$$D_K(C) = M$$

Symmetric algorithms can be divided into two categories. Some operate on the plaintext a single bit (or sometimes byte) at a time; these are called **stream algorithms** or **stream ciphers**. Others operate on the plaintext in groups of bits. The groups of bits are called **blocks**, and the algorithms are called **block algorithms** or **block ciphers**. For modern computer algorithms, a typical block size is 64 bits—large enough to preclude analysis and small enough to be workable. (Before computers, algorithms generally operated on plaintext one character at a time. You can think of this as a stream algorithm operating on a stream of characters.)

Public-Key Algorithms

Public-key algorithms (also called asymmetric algorithms) are designed so that the key used for encryption is different from the key used for decryption. Furthermore, the decryption key cannot (at least in any reasonable amount of time) be calculated from the encryption key. The algorithms are called "public-key" because the encryption key can be made public: A complete stranger can use the encryption key to encrypt a message, but only a specific person with the corresponding decryp-

tion key can decrypt the message. In these systems, the encryption key is often called the **public key**, and the decryption key is often called the **private key**. The private key is sometimes also called the secret key, but to avoid confusion with symmetric algorithms, that tag won't be used here.

Encryption using public key K is denoted by:

$$E_K(M) = C$$

Even though the public key and private key are different, decryption with the corresponding private key is denoted by:

$$D_K(C) = M$$

Sometimes, messages will be encrypted with the private key and decrypted with the public key; this is used in digital signatures (see Section 2.6). Despite the possible confusion, these operations are denoted by, respectively:

$$E_K(M) = C$$
$$D_K(C) = M$$

Cryptanalysis

The whole point of cryptography is to keep the plaintext (or the key, or both) secret from eavesdroppers (also called adversaries, attackers, interceptors, interlopers, intruders, opponents, or simply the enemy). Eavesdroppers are assumed to have complete access to the communications between the sender and receiver.

Cryptanalysis is the science of recovering the plaintext of a message without access to the key. Successful cryptanalysis may recover the plaintext or the key. It also may find weaknesses in a cryptosystem that eventually lead to the previous results. (The loss of a key through noncryptanalytic means is called a **compromise**.)

An attempted cryptanalysis is called an **attack**. A fundamental assumption in cryptanalysis, first enunciated by the Dutchman A. Kerckhoffs in the nineteenth century, is that the secrecy must reside entirely in the key [794]. Kerckhoffs assumes that the cryptanalyst has complete details of the cryptographic algorithm and implementation. (Of course, one would assume that the CIA does not make a habit of telling Mossad about its cryptographic algorithms, but Mossad probably finds out anyway.) While real-world cryptanalysts don't always have such detailed information, it's a good assumption to make. If others can't break an algorithm, even with knowledge of how it works, then they certainly won't be able to break it without that knowledge.

There are four general types of cryptanalytic attacks. Of course, each of them assumes that the cryptanalyst has complete knowledge of the encryption algorithm used:

1. **Ciphertext-only attack**. The cryptanalyst has the ciphertext of several messages, all of which have been encrypted using the same encryption algorithm. The cryptanalyst's job is to recover the plaintext of as many messages as possible, or better yet to deduce the key (or keys) used to

Think of it as a way of fingerprinting files. If you want to verify that someone has a particular file (that you also have), but you don't want him to send it to you, then ask him for the hash value. If he sends you the correct hash value, then it is almost certain that he has that file. This is particularly useful in financial transactions, where you don't want a withdrawal of $100 to turn into a withdrawal of $1000 somewhere in the network. Normally, you would use a one-way hash function without a key, so that anyone can verify the hash. If you want only the recipient to be able to verify the hash, then read the next section.

Message Authentication Codes

A **message authentication code** (MAC), also known as a data authentication code (DAC), is a one-way hash function with the addition of a secret key (see Section 18.14). The hash value is a function of both the pre-image and the key. The theory is exactly the same as hash functions, except only someone with the key can verify the hash value. You can create a MAC out of a hash function or a block encryption algorithm; there are also dedicated MACs.

2.5 Communications Using Public-Key Cryptography

Think of a symmetric algorithm as a safe. The key is the combination. Someone with the combination can open the safe, put a document inside, and close it again. Someone else with the combination can open the safe and take the document out. Anyone without the combination is forced to learn safecracking.

In 1976, Whitfield Diffie and Martin Hellman changed that paradigm of cryptography forever [496]. (The NSA has claimed knowledge of the concept as early as 1966, but has offered no proof.) They described **public-key cryptography**. They used two different keys—one public and the other private. It is computationally hard to deduce the private key from the public key. Anyone with the public key can encrypt a message but not decrypt it. Only the person with the private key can decrypt the message. It is as if someone turned the cryptographic safe into a mailbox. Putting mail in the mailbox is analogous to encrypting with the public key; anyone can do it. Just open the slot and drop it in. Getting mail out of a mailbox is analogous to decrypting with the private key. Generally it's hard; you need welding torches. However, if you have the secret (the physical key to the mailbox), it's easy to get mail out of a mailbox.

Mathematically, the process is based on the trap-door one-way functions previously discussed. Encryption is the easy direction. Instructions for encryption are the public key; anyone can encrypt a message. Decryption is the hard direction. It's made hard enough that people with Cray computers and thousands (even millions) of years couldn't decrypt the message without the secret. The secret, or trapdoor, is the private key. With that secret, decryption is as easy as encryption.

This is how Alice can send a message to Bob using public-key cryptography:

(1) Alice and Bob agree on a public-key cryptosystem.

(2) Bob sends Alice his public key.
(3) Alice encrypts her message using Bob's public key and sends it to Bob.
(4) Bob decrypts Alice's message using his private key.

Notice how public-key cryptography solves the key-management problem with symmetric cryptosystems. Before, Alice and Bob had to agree on a key in secret. Alice could choose one at random, but she still had to get it to Bob. She could hand it to him sometime beforehand, but that requires foresight. She could send it to him by secure courier, but that takes time. Public-key cryptography makes it easy. With no prior arrangements, Alice can send a secure message to Bob. Eve, listening in on the entire exchange, has Bob's public key and a message encrypted in that key, but cannot recover either Bob's private key or the message.

More commonly, a network of users agrees on a public-key cryptosystem. Every user has his or her own public key and private key, and the public keys are all published in a database somewhere. Now the protocol is even easier:

(1) Alice gets Bob's public key from the database.
(2) Alice encrypts her message using Bob's public key and sends it to Bob.
(3) Bob then decrypts Alice's message using his private key.

In the first protocol, Bob had to send Alice his public key before she could send him a message. The second protocol is more like traditional mail. Bob is not involved in the protocol until he wants to read his message.

Hybrid Cryptosystems

The first public-key algorithms became public at the same time that DES was being discussed as a proposed standard. This resulted in some partisan politics in the cryptographic community. As Diffie described it [494]:

> The excitement public key cryptosystems provoked in the popular and scientific press was not matched by corresponding acceptance in the cryptographic establishment, however. In the same year that public key cryptography was discovered, the National Security Agency (NSA), proposed a conventional cryptographic system, designed by International Business Machines (IBM), as a federal *Data Encryption Standard* (DES). Marty Hellman and I criticized the proposal on the ground that its key was too small, but manufacturers were gearing up to support the proposed standard and our criticism was seen by many as an attempt to disrupt the standards-making process to the advantage of our own work. Public key cryptography in its turn was attacked, in sales literature [1125] and technical papers [849,1159] alike, more as though it were a competing product than a recent research discovery. This, however, did not deter the NSA from claiming its share of the credit. Its director, in the words of the *Encyclopedia Britannica* [1461], pointed out that "two-key cryptography had been discovered at the agency a decade earlier," although no evidence for this claim was ever offered publicly.

In the real world, public-key algorithms are not a substitute for symmetric algorithms. They are not used to encrypt messages; they are used to encrypt keys. There are two reasons for this:

1. Public-key algorithms are slow. Symmetric algorithms are generally at least 1000 times faster than public-key algorithms. Yes, computers are getting faster and faster, and in 15 years computers will be able to do public-key cryptography at speeds comparable to symmetric cryptography today. But bandwidth requirements are also increasing, and there will always be the need to encrypt data faster than public-key cryptography can manage.
2. Public-key cryptosystems are vulnerable to chosen-plaintext attacks. If $C = E(P)$, when P is one plaintext out of a set of n possible plaintexts, then a cryptanalyst only has to encrypt all n possible plaintexts and compare the results with C (remember, the encryption key is public). He won't be able to recover the decryption key this way, but he will be able to determine P.

A chosen-plaintext attack can be particularly effective if there are relatively few possible encrypted messages. For example, if P were a dollar amount less than $1,000,000, this attack would work; the cryptanalyst tries all million possible dollar amounts. (Probabilistic encryption solves the problem; see Section 23.15.) Even if P is not as well-defined, this attack can be very effective. Simply knowing that a ciphertext does not correspond to a particular plaintext can be useful information. Symmetric cryptosystems are not vulnerable to this attack because a cryptanalyst cannot perform trial encryptions with an unknown key.

In most practical implementations public-key cryptography is used to secure and distribute **session keys**; those session keys are used with symmetric algorithms to secure message traffic [879]. This is sometimes called a **hybrid cryptosystem**.

(1) Bob sends Alice his public key.

(2) Alice generates a random session key, K, encrypts it using Bob's public key, and sends it to Bob.

$$E_B(K)$$

(3) Bob decrypts Alice's message using his private key to recover the session key.

$$D_B(E_B(K)) = K$$

(4) Both of them encrypt their communications using the same session key.

Using public-key cryptography for key distribution solves a very important key-management problem. With symmetric cryptography, the data encryption key sits around until it is used. If Eve ever gets her hands on it, she can decrypt messages encrypted with it. With the previous protocol, the session key is created when it is needed to encrypt communications and destroyed when it is no longer needed. This drastically reduces the risk of compromising the session key. Of course, the private

key is vulnerable to compromise, but it is at less risk because it is only used once per communication to encrypt a session key. This is further discussed in Section 3.1.

Merkle's Puzzles

Ralph Merkle invented the first construction of public-key cryptography. In 1974 he registered for a course in computer security at the University of California, Berkeley, taught by Lance Hoffman. His term paper topic, submitted early in the term, addressed the problem of "Secure Communication over Insecure Channels" [1064]. Hoffman could not understand Merkle's proposal and eventually Merkle dropped the course. He continued to work on the problem, despite continuing failure to make his results understood.

Merkle's technique was based on "puzzles" that were easier to solve for the sender and receiver than for an eavesdropper. Here's how Alice sends an encrypted message to Bob without first having to exchange a key with him.

(1) Bob generates 2^{20}, or about a million, messages of the form: "This is puzzle number x. This is the secret key number y," where x is a random number and y is a random secret key. Both x and y are different for each message. Using a symmetric algorithm, he encrypts each message with a different 20-bit key and sends them all to Alice.

(2) Alice chooses one message at random and performs a brute-force attack to recover the plaintext. This is a large, but not impossible, amount of work.

(3) Alice encrypts her secret message with the key she recovered and some symmetric algorithm, and sends it to Bob along with x.

(4) Bob knows which secret key y he encrypts in message x, so he can decrypt the message.

Eve can break this system, but she has to do far more work than either Alice or Bob. To recover the message in step (3), she has to perform a brute-force attack against each of Bob's 2^{20} messages in step (1); this attack has a complexity of 2^{40}. The x values won't help Eve either; they were assigned randomly in step (1). In general, Eve has to expend approximately the square of the effort that Alice expends.

This n to n^2 advantage is small by cryptographic standards, but in some circumstances it may be enough. If Alice and Bob can try ten thousand keys per second, it will take them a minute each to perform their steps and another minute to communicate the puzzles from Bob to Alice on a 1.544 MB link. If Eve had comparable computing facilities, it would take her about a year to break the system. Other algorithms are even harder to break.

2.6 DIGITAL SIGNATURES

Handwritten signatures have long been used as proof of authorship of, or at least agreement with, the contents of a document. What is it about a signature that is so compelling [1392]?

each of the users; he can read all past communications traffic that he has saved, and all future communications traffic. All he has to do is to tap the communications lines and listen to the encrypted message traffic.

The other problem with this system is that Trent is a potential bottleneck. He has to be involved in every key exchange. If Trent fails, that disrupts the entire system.

Key Exchange with Public-Key Cryptography

The basic hybrid cryptosystem was discussed in Section 2.5. Alice and Bob use public-key cryptography to agree on a session key, and use that session key to encrypt data. In some practical implementations, both Alice's and Bob's signed public keys will be available on a database. This makes the key-exchange protocol even easier, and Alice can send a secure message to Bob even if he has never heard of her:

(1) Alice gets Bob's public key from the KDC.

(2) Alice generates a random session key, encrypts it using Bob's public key, and sends it to Bob.

(3) Bob then decrypts Alice's message using his private key.

(4) Both of them encrypt their communications using the same session key.

Man-in-the-Middle Attack

While Eve cannot do better than try to break the public-key algorithm or attempt a ciphertext-only attack on the ciphertext, Mallory is a lot more powerful than Eve. Not only can he listen to messages between Alice and Bob, he can also modify messages, delete messages, and generate totally new ones. Mallory can imitate Bob when talking to Alice and imitate Alice when talking to Bob. Here's how the attack works:

(1) Alice sends Bob her public key. Mallory intercepts this key and sends Bob his own public key.

(2) Bob sends Alice his public key. Mallory intercepts this key and sends Alice his own public key.

(3) When Alice sends a message to Bob, encrypted in "Bob's" public key, Mallory intercepts it. Since the message is really encrypted with his own public key, he decrypts it with his private key, re-encrypts it with Bob's public key, and sends it on to Bob.

(4) When Bob sends a message to Alice, encrypted in "Alice's" public key, Mallory intercepts it. Since the message is really encrypted with his own public key, he decrypts it with his private key, re-encrypts it with Alice's public key, and sends it on to Alice.

Even if Alice's and Bob's public keys are stored on a database, this attack will work. Mallory can intercept Alice's database inquiry and substitute his own public

CHAPTER 22

Key-Exchange Algorithms

22.1 DIFFIE-HELLMAN

Diffie-Hellman was the first public-key algorithm ever invented, way back in 1976 [496]. It gets its security from the difficulty of calculating discrete logarithms in a finite field, as compared with the ease of calculating exponentiation in the same field. Diffie-Hellman can be used for key distribution—Alice and Bob can use this algorithm to generate a secret key—but it cannot be used to encrypt and decrypt messages.

The math is simple. First, Alice and Bob agree on a large prime, n and g, such that g is primitive mod n. These two integers don't have to be secret; Alice and Bob can agree to them over some insecure channel. They can even be common among a group of users. It doesn't matter.

Then, the protocol goes as follows:

(1) Alice chooses a random large integer x and sends Bob

$$X = g^x \bmod n$$

(2) Bob chooses a random large integer y and sends Alice

$$Y = g^y \bmod n$$

(3) Alice computes

$$k = Y^x \bmod n$$

(4) Bob computes

$$k' = X^y \bmod n$$

Both k and k' are equal to $g^{xy} \bmod n$. No one listening on the channel can compute that value; they only know n, g, X, and Y. Unless they can compute the discrete log-

arithm and recover x or y, they do not solve the problem. So, k is the secret key that both Alice and Bob computed independently.

The choice of g and n can have a substantial impact on the security of this system. The number $(n - 1)/2$ should also be a prime [1253]. And most important, n should be large: The security of the system is based on the difficulty of factoring numbers the same size as n. You can choose any g, such that g is primitive mod n; there's no reason not to choose the smallest g you can—generally a one-digit number. (And actually, g does not have to be primitive; it just has to generate a large subgroup of the multiplicitive group mod n.)

Diffie-Hellman with Three or More Parties

The Diffie-Hellman key-exchange protocol can easily be extended to work with three or more people. In this example, Alice, Bob, and Carol together generate a secret key.

(1) Alice chooses a random large integer x and sends Bob

$$X = g^x \bmod n$$

(2) Bob chooses a random large integer y and sends Carol

$$Y = g^y \bmod n$$

(3) Carol chooses a random large integer z and sends Alice

$$Z = g^z \bmod n$$

(4) Alice sends Bob

$$Z' = Z^x \bmod n$$

(5) Bob sends Carol

$$X' = X^y \bmod n$$

(6) Carol sends Alice

$$Y' = Y^z \bmod n$$

(7) Alice computes

$$k = Y'^x \bmod n$$

(8) Bob computes

$$k = Z'^y \bmod n$$

(9) Carol computes

$$k = X'^z \bmod n$$

The secret key, k, is equal to $g^{xyz} \bmod n$, and no one else listening in on the communications can compute that value. The protocol can be easily extended to four or more people; just add more people and more rounds of computation.

Extended Diffie-Hellman

Diffie-Hellman also works in commutitive rings [1253]. Z. Shmuley and Kevin McCurley studied a variant of the algorithm where the modulus is a composite number [1442,1038]. V. S. Miller and Neal Koblitz extended this algorithm to elliptic curves [1095,867]. Taher ElGamal used the basic idea to develop an encryption and digital signature algorithm (see Section 19.6).

This algorithm also works in the Galois field $GF(2^k)$ [1442,1038]. Some implementations take this approach [884,1631,1632], because the computation is much quicker. Similarly, cryptanalytic computation is equally fast, so it is important to carefully choose a field large enough to ensure security.

Hughes

This variant of Diffie-Hellman allows Alice to generate a key and send it to Bob [745].

(1) Alice chooses a random large integer x and generates

$$k = g^x \bmod n$$

(2) Bob chooses a random large integer y and sends Alice

$$Y = g^y \bmod n$$

(3) Alice sends Bob

$$X = Y^x \bmod n$$

(4) Bob computes

$$z = y^{-1}$$
$$k' = X^z \bmod n$$

If everything goes correctly, $k = k'$.

The advantage of this protocol over Diffie-Hellman is that k can be computed before any interaction, and Alice can encrypt a message using k prior to contacting Bob. She can send it to a variety of people and interact with them to exchange the key individually later.

Key Exchange Without Exchanging Keys

If you have a community of users, each could publish a public key, $X = g^x \bmod n$, in a common database. If Alice wants to communicate with Bob, she just has to retrieve Bob's public key and generate their shared secret key. She could then encrypt a message with that key and send it to Bob. Bob would retrieve Alice's public key to generate the shared secret key.

Each pair of users would have a unique secret key, and no prior communication between users is required. The public keys have to be certified to prevent spoofing attacks and should be changed regularly, but otherwise this is a pretty clever idea.

Patents

The Diffie-Hellman key-exchange algorithm is patented in the United States [718] and Canada [719]. A group called Public Key Partners (PKP) licenses the patent, along with other public-key cryptography patents (see Section 25.5). The U.S. patent will expire on April 29, 1997.

22.2 Station-to-Station Protocol

Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack. One way to prevent this problem is to have Alice and Bob sign their messages to each other [500].

This protocol assumes that Alice has a certificate with Bob's public key and that Bob has a certificate with Alice's public key. These certificates have been signed by some trusted authority outside this protocol. Here's how Alice and Bob generate a secret key, k.

(1) Alice generates a random number, x, and sends it to Bob.

(2) Bob generates a random number, y. Using the Diffie-Hellman protocol he computes their shared key based on x and y: k. He signs x and y, and encrypts the signature using k. He then sends that, along with y, to Alice.

$$y, E_k[S_B[x,y]]$$

(3) Alice also computes k. She decrypts the rest of Bob's message and verifies his signature. Then she sends Bob a signed message consisting of x and y, encrypted in their shared key.

$$E_k[S_A[x,y]]$$

(4) Bob decrypts the message and verifies Alice's signature.

22.3 Shamir's Three-Pass Protocol

This protocol, invented by Adi Shamir but never published, enables Alice and Bob to communicate securely without any advance exchange of either secret keys or public keys [1008].

This assumes the existence of a symmetric cipher that is commutative, that is:

$$E_A[E_B[P]] = E_B[E_A[P]]$$

Alice's secret key is A; Bob's secret key is B. Alice wants to send a message, M, to Bob. Here's the protocol.

(1) Alice encrypts M with her key and sends Bob

$$C_1 = E_A[M]$$

(2) Bob encrypts C_1 with his key and sends Alice

$$C_2 = E_B[E_A[M]]$$